OnlineRePair: A Recompressor for XML Structures

OnlineRePair: a Recompressor for XML Structures

Stefan Böttcher*, Rita Hartel*, Thomas Jacobs*, and Sebastian Maneth†

*Universität Paderborn
Fürstenallee 11
33102 Paderborn, Germany
{stb@,rst@,tjacobs@mail.}upb.de

†University of Edinburgh
10 Crichton Street
Edinburgh, EH8 9AB, UK
smaneth@inf.ed.ac.uk

Grammar-based compression yields high compression ratios for XML document trees. The best known compressor, TreeRePair [1] compresses typical XML tree structures to about 3% of their original size. It does so by factoring out repeated connected subgraphs of the tree. The result is a *straight-line context-free tree* (*SLT*) grammar. Certain procedures, such as Core XPath query evaluation or equivalence check, can be carried out directly on an SLT grammar (without prior decompression), thus possibly producing a speed-up.

Despite these successes in grammar-based compression, one of the holy grails has been its use as a mutable data structure. If a compressed tree is manipulated, for instance by repeated update operations, then the compression ratio rapidly degrades. The best known method to deal with this, has been to periodically decompress the tree and run TreeRePair from scratch. This is highly problematic, as it may take exponential time and space. Here we present the first implementation of mutable compressed trees: our *OnlineRePair* [2] algorithm takes as input an SLT grammar G, and in polynomial time produces a new (smaller) grammar G'. Intuitively, the grammar G' is obtained by running the Repair compression algorithm over G. This is non-trivial because the basic step of Repair, namely the replacement of a digram by a nonterminal, is challenging to implement efficiently over SLT grammars: a digram (an edge together with its two nodes) can span over several grammar rules. Thus, these rules need to be applied to obtain the digram. The crucial step was to find a minimal number of rules and an economic way of rule application that avoids decompression.

Our experimental evaluation shows that (*i*) the SLT grammars produced by OnlineRePair are *at least* as small as those obtained by decompression followed by Repair compression, and (*ii*) in terms of run time, OnlineRePair outperforms the decompress-compress approach; for the largest files, OnlineRePair even outperforms the sole compression time of TreeRePair without considering the additional decompression required prior to the compression.

References

[1] Markus Lohrey, Sebastian Maneth, and Roy Mennicke, "XML tree structure compression using RePair," *Inf. Syst.*, vol. 38, pp. 1150–1167, 2013.

[2] Thomas Jacobs, "Multi-version Grammar-based XML Compression," M.S. thesis, University of Paderborn, October 2014.